A Priority Model for Named Entities

Lorraine Tanabe
National Center for Biotechnology Information
Bethesda, MD 20894
tanabe@ncbi.nlm.nih.gov

W. John Wilbur
National Center for Biotechnology Information
Bethesda, MD 20894
wilbur@ncbi.nlm.nih.gov

Abstract

We introduce a new approach to named entity classification which we term a Priority Model. We also describe the construction of a semantic database called SemCat consisting of a large number of semantically categorized names relevant to biomedicine. We used SemCat as training data to investigate name classification techniques. We generated a statistical language model and probabilistic context-free grammars for gene and protein name classification, and compared the results with the new model. For all three methods, we used a variable order Markov model to predict the nature of strings not represented in the training data. The Priority Model achieves an F-measure of 0.958-0.960, consistently higher than the statistical language model and probabilistic context-free grammar.

1 Introduction

Automatic recognition of gene and protein names is a challenging first step towards text mining the biomedical literature. Advances in the area of gene and protein named entity recognition (NER) have been accelerated by freely available tagged corpora (Kim et al., 2003, Cohen et al., 2005, Smith et al., 2005, Tanabe et al., 2005). Such corpora have made it possible for standardized evaluations such as Task 1A of the first BioCreative Workshop (Yeh et al., 2005).

Although state-of-the-art systems now perform at the level of 80-83% F-measure, this is still well below the range of 90-97% for non-biomedical NER. The main reasons for this performance disparity are 1) the complexity of the genetic nomenclature and 2) the confusion of gene and protein names with other biomedical entities, as well as with common English words. In an effort to alleviate the confusion with other biomedical entities we have assembled a database consisting of named entities appearing in the literature of biomedicine together with information on their ontological categories. We use this information in an effort to better understand how to classify names as representing genes/proteins or not.

2 Background

A successful gene and protein NER system must address the complexity and ambiguity inherent in this domain. Hand-crafted rules alone are unable to capture these phenomena in large biomedical text collections. Most biomedical NER systems use some form of language modeling, consisting of an observed sequence of words and a hidden sequence of tags. The goal is to find the tag sequence with maximal probability given the observed word sequence. McDonald and Pereira (2005) use conditional random fields (CRF) to identify the beginning, inside and outside of gene and protein names. GuoDong et al. (2005) use an ensemble of one support vector machine and two Hidden Markov Models (HMMs). Kinoshita et al. (2005) use a second-order Markov model. Dingare et al. (2005) use a maximum entropy Markov model (MEMM) with large feature sets.

Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06, pages 33–40,
New York City, June 2006.

NER is a difficult task because it requires both the identification of the boundaries of an entity in text, and the classification of that entity. In this paper, we focus on the classification step. Spasic et al. (2005) use the MaSTerClass case-based reasoning system for biomedical term classification. MaSTerClass uses term contexts from an annotated corpus of 2072 MEDLINE abstracts related to *nuclear receptors* as a basis for classifying new terms. Its set of classes is a subset of the UMLS Semantic Network (McCray, 1989), that does not include genes and proteins. Liu et al. (2002) classified terms that represent multiple UMLS concepts by examining the *conceptual relatives* of the concepts. Hatzivassiloglou et al. (2001) classified terms known to belong to the classes *Protein*, *Gene* and/or *RNA* using unsupervised learning, achieving accuracy rates up to 85%. The AZuRE system (Podowski et al., 2004) uses a separate modified Naive Bayes model for each of 20K genes. A term is disambiguated based on its contextual similarity to each model. Nenadic et al. (2003) recognized the importance of terminological knowledge for biomedical text mining. They used the C/NC-methods, calculating both the intrinsic characteristics of terms (such as their frequency of occurrence as substrings of other terms), and the context of terms as linear combinations. These biomedical classification systems all rely on the context surrounding named entities. While we recognize the importance of context, we believe one must strive for the appropriate blend of information coming from the context and information that is inherent in the name itself. This explains our focus on names without context in this work.

We believe one can improve gene and protein entity classification by using more training data and/or using a more appropriate model for names. Current sources of training data are deficient in important biomedical terminologies like cell line names. To address this deficiency, we constructed the SemCat database, based on a subset of the UMLS Semantic Network enriched with categories from the GENIA Ontology (Kim et al, 2003), and a few new semantic types. We have populated SemCat with over 5 million entities of interest from

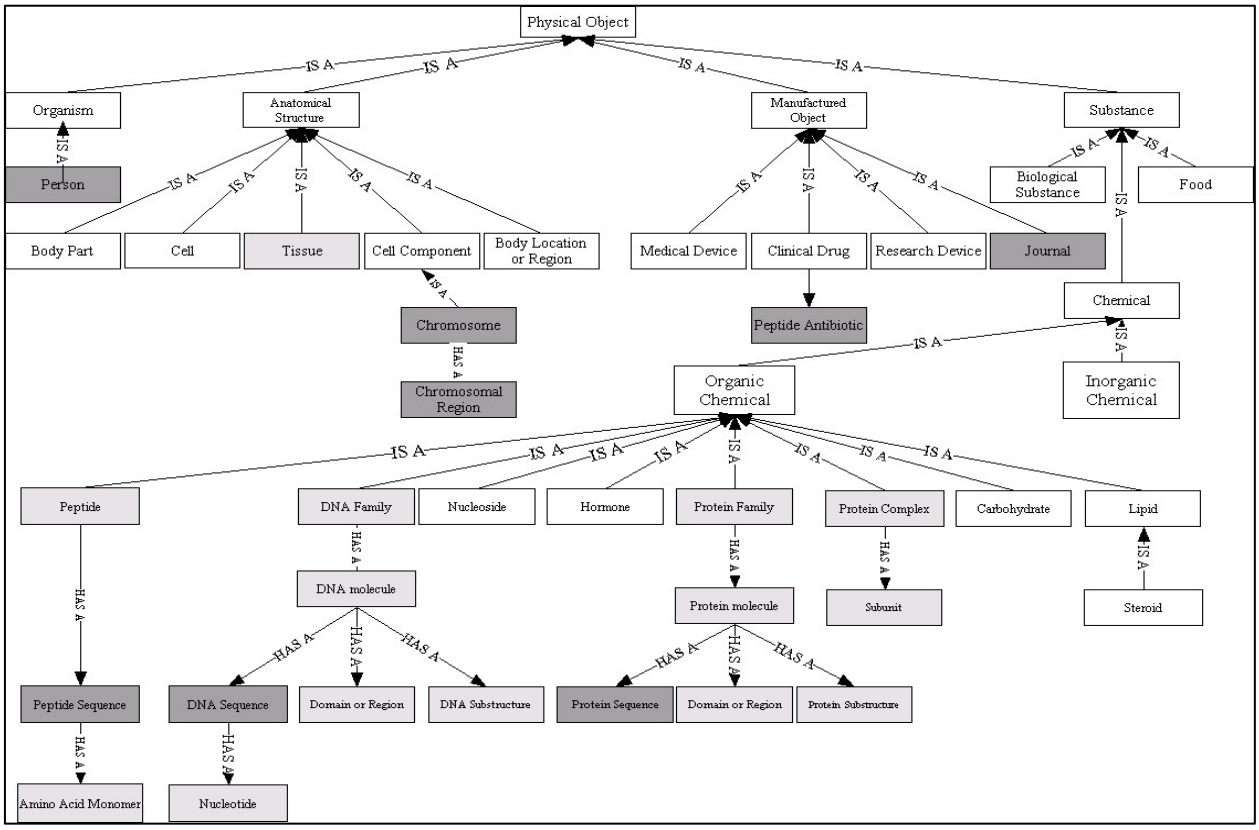

Figure 1. SemCat Physical Object Hierarchy. White = UMLS SN, Light Grey = GENIA semantic types, Dark Grey = New semantic types.

standard knowledge sources like the UMLS (Lindberg et al., 1993), the Gene Ontology (GO) (The Gene Ontology Consortium, 2000), Entrez Gene (Maglott et al., 2005), and GENIA, as well as from the World Wide Web. In this paper, we use SemCat data to compare three probabilistic frameworks for named entity classification.

3 Methods

We constructed the SemCat database of biomedical entities, and used these entities to train and test three probabilistic approaches to gene and protein name classification: 1) a statistical language model with Witten-Bell smoothing, 2) probabilistic context-free grammars (PCFGs) and 3) a new approach we call a Priority Model for named entities. As one component in all of our classification algorithms we use a variable order Markov Model for strings.

3.1 SemCat Database Construction

The UMLS Semantic Network (SN) is an ongoing project at the National Library of Medicine. Many users have modified the SN for their own research domains. For example, Yu et al. (1999) found that the SN was missing critical components in the genomics domain, and added six new semantic types including *Protein Structure* and *Chemical Complex*. We found that a subset of the SN would be sufficient for gene and protein name classification, and added some new semantic types for better coverage. We shifted some semantic types from suboptimal nodes to ones that made more sense from a genomics standpoint. For example, there were two problems with *Gene or Genome*. Firstly, genes and genomes are not synonymous, and secondly, placement under the semantic type *Fully Formed Anatomical Structure* is suboptimal from a genomics perspective. Since a gene in this context is better understood as an organic chemical, we deleted *Gene or Genome*, and added the GENIA semantic types for genomics entities under *Organic Chemical*. The SemCat Physical Object hierarchy is shown in Figure 1. Similar hierarchies exist for the SN Conceptual Entity and Event trees. A number of the categories have been supplemented with automatically extracted entities from MEDLINE, derived from regular expression pattern matching. Currently, SemCat has 77 semantic types, and 5.11M non-unique entries. Additional entities from MEDLINE are being manually classified via an annotation website. Unlike the Termino database (Harkema et al. (2004), which contains terminology annotated with morpho-syntactic and conceptual information, SemCat currently consists of gazetteer lists only.

For our experiments, we generated two sets of training data from SemCat, Gene-Protein (*GP*) and Not-Gene-Protein (*NGP*). *GP* consists of specific terms from the semantic types DNA MOLECULE, PROTEIN MOLECULE, DNA FAMILY, PROTEIN FAMILY, PROTEIN COMPLEX and PROTEIN SUBUNIT. *NGP* consists of entities from all other SemCat types, along with generic entities from the *GP* semantic types. Generic entities were automatically eliminated from *GP* using pattern matching to manually tagged generic phrases like *abnormal protein*, *acid domain*, and *RNA*.

Many SemCat entries contain commas and parentheses, for example, "*receptors, tgf beta.*" A better form for natural language processing would be "*tgf beta receptors*." To address this problem, we automatically generated variants of phrases in *GP* with commas and parentheses, and found their counts in MEDLINE. We empirically determined the heuristic rule of replacing the phrase with its second most frequent variant, based on the observation that the most frequent variant is often too generic. For example, the following are the phrase variant counts for "*heat shock protein (dnaj)":*

- heat shock protein (dnaj) 0
- dnaj heat shock protein 84
- heat shock protein 122954
- heat shock protein dnaj 41

Thus, the phrase kept for *GP* is *dnaj heat shock protein*.

After purifying the sets and removing ambiguous full phrases (ambiguous words were retained), *GP* contained 1,001,188 phrases, and *NGP* contained 2,964,271 phrases. From these, we randomly generated three train/test divisions of 90% train/10% test (*gp1*, *gp2*, *gp3*), for the evaluation.

3.2 Variable Order Markov Model for Strings

As one component in our classification algorithms we use a variable order Markov Model for strings. Suppose *C* represents a class and $x_1 x_2 x_3 ... x_n$ repre-

sents a string of characters. In order to estimate the probability that $x_1x_2x_3...x_n$ belongs to C we apply Bayes' Theorem to write

$$p(C \mid x_1x_2x_3...x_n) = \frac{p(x_1x_2x_3...x_n \mid C)\, p(C)}{p(x_1x_2x_3...x_n)} \quad (1)$$

Because $p(x_1x_2x_3...x_n)$ does not depend on the class and because we are generally comparing probability estimates between classes, we ignore this factor in our calculations and concentrate our efforts on evaluating $p(x_1x_2x_3...x_n \mid C)\, p(C)$. First we write

$$p(x_1x_2x_3...x_n \mid C) = \prod_{k=1}^{n} p(x_k \mid x_1x_2x_3...x_{k-1}, C) \quad (2)$$

which is an exact equality. The final step is to give our best approximation to each of the numbers $p(x_k \mid x_1x_2x_3...x_{k-1}, C)$. To make these approximations we assume that we are given a set of strings and associated probabilities $\{(s_i, p_i)\}_{i=1}^{M}$ where for each i, $p_i > 0$ and p_i is assumed to represent the probability that s_i belongs to the class C. Then for the given string $x_1x_2x_3...x_n$ and a given k we let $r \geq 1$ be the smallest integer for which $x_rx_{r+1}x_{r+2}...x_k$ is a contiguous substring in at least one of the strings s_i. Now let N' be the set of all i for which $x_rx_{r+1}x_{r+2}...x_k$ is a substring of s_i and let N be the set of all i for which $x_rx_{r+1}x_{r+2}...x_{k-1}$ is a substring of s_i. We set

$$p(x_k \mid x_1x_2x_3...x_{k-1}, C) = \frac{\sum_{i \in N'} p_i}{\sum_{i \in N} p_i}. \quad (3)$$

In some cases it is appropriate to assume that $p(C)$ is proportional to $\sum_{i=1}^{M} p_i$ or there may be other ways to make this estimate. This basic scheme works well, but we have found that we can obtain a modest improvement by adding a unique start character to the beginning of each string. This character is assumed to occur nowhere else but as the first character in all strings dealt with including any string whose probability we are estimating. This forces the estimates of probabilities near the beginnings of strings to come from estimates based on the beginnings of strings. We use this approach in all of our classification algorithms.

Table 1. Each fragment in the left column appears in the training data and the probability in the right column represents the probability of seeing the underlined portion of the string given the occurrence of the initial ununderlined portion of the string in a training string.

GP	
!apoe	9.55×10^{-7}
oe-e	2.09×10^{-3}
e-epsilon	4.00×10^{-2}
$p(apoe-epsilon \mid GP)$	7.98×10^{-11}
$p(GP \mid apoe-epsilon)$	0.98448
NGP	
!apoe	8.88×10^{-8}
poe-	1.21×10^{-2}
oe-e	6.10×10^{-2}
e-epsilon	6.49×10^{-3}
$p(apoe-epsilon \mid NGP)$	4.25×10^{-13}
$p(NGP \mid apoe-epsilon)$	0.01552

In Table 1, we give an illustrative example of the string apoe-epsilon which does not appear in the training data. A PubMed search for apoe-epsilon gene returns 269 hits showing the name is known. But it does not appear in this exact form in SemCat.

3.3 Language Model with Witten-Bell Smoothing

A statistical *n*-gram model is challenged when a bigram in the test set is absent from the training set, an unavoidable situation in natural language due to Zipf's law. Therefore, some method for assigning nonzero probability to novel *n*-grams is required. For our language model (LM), we used Witten-Bell smoothing, which reserves probability mass for out of vocabulary values (Witten and Bell, 1991, Chen and Goodman, 1998). The discounted probability is calculated as

$$\hat{P}(w_{i-n+1}...w_{i-1}) = \frac{\#(w_{i-n+1}...w_i)}{\#(w_{i-n+1}...w_{i-1}) + D(w_{i-n+1}...w_{i-1})} \quad (4)$$

where $D(w_{i-n+1}...w_{i-1})$ is the number of distinct words that can appear after $w_{i-n+1}...w_{i-1}$ in the training data. Actual values assigned to tokens outside the training data are not assigned uniformly but are filled in using a variable order Markov Model based on the strings seen in the training data.

3.4 Probabilistic Context-Free Grammar

The Probabilistic Context-Free Grammar (PCFG) or Stochastic Context-Free Grammar (SCFG) was originally formulated by Booth (1969). For technical details we refer the reader to Charniak (1993). For gene and protein name classification, we tried two different approaches. In the first PCFG method (PCFG-3), we used the following simple productions:

1) CATP → CATP CATP
2) CATP → CATP postCATP
3) CATP → preCATP CATP

CATP refers to the category of the phrase, *GP* or *NGP*. The prefixes *pre* and *post* refer to beginnings and endings of the respective strings. We trained two separate grammars, one for the positive examples, *GP*, and one for the negative examples, *NGP*. Test cases were tagged based on their score from each of the two grammars.

In the second PCFG method (PCFG-8), we combined the positive and negative training examples into one grammar. The minimum number of non-terminals necessary to cover the training sets *gp1-3* was six {CATP, preCATP, postCATP, NotCATP, preNotCATP, postNotCATP}. CATP represents a string from *GP*, and NotCATP represents a string from *NGP*. We used the following production rules:

1) CATP → CATP CATP
2) CATP → CATP postCATP
3) CATP → preCATP CATP
4) CATP → NotCATP CATP
5) NotCATP → NotCATP NotCATP
6) NotCATP → NotCATP postNotCATP
7) NotCATP→ preNotCATP NotCATP
8) NotCATP → CATP NotCATP

It can be seen that (4) is necessary for strings like "human p53," and (8) covers strings like "p53 pathway."

In order to deal with tokens that do not appear in the training data we use variable order Markov Models for strings. First the grammar is trained on the training set of names. Then any token appearing in the training data will have assigned to it the tags appearing on the right side of any rule of the grammar (essentially part-of-speech tags) with probabilities that are a product of the training. We then construct a variable order Markov Model for each tag type based on the tokens in the training data and the assigned probabilities for that tag type. These Models (three for PCFG-3 and six for PCFG-8) are then used to assign the basic tags of the grammar to any token not seen in training. In this way the grammars can be used to classify any name even if its tokens are not in the training data.

3.5 Priority Model

There are problems with the previous approaches when applied to names. For example, suppose one is dealing with the name "human liver alkaline phosphatase" and class C_1 represents protein names and class C_2 anatomical names. In that case a language model is no more likely to favor C_1 than C_2. We have experimented with PCFGs and have found the biggest challenge to be how to choose the grammar. After a number of attempts we have still found problems of the "human liver alkaline phosphatase" type to persist.

The difficulties we have experienced with language models and PCFGs have led us to try a different approach to model named entities. As a general rule in a phrase representing a named entity a word to the right is more likely to be the head word or the word determining the nature of the entity than a word to the left. We follow this rule and construct a model which we will call a Priority Model. Let T_1 be the set of training data (names) for class C_1 and likewise T_2 for C_2. Let $\{t_\alpha\}_{\alpha \in A}$ denote the set of all tokens used in names contained in $T_1 \cup T_2$. Then for each token t_α, $\alpha \in A$, we assume there are associated two probabilities p_α and q_α with the interpretation that p_α is the

probability that the appearance of the token t_α in a name indicates that name belongs to class C_1 and q_α is the probability that t_α is a reliable indicator of the class of a name. Let $n = t_{\alpha(1)} t_{\alpha(2)} \ldots t_{\alpha(k)}$ be composed of the tokens on the right in the given order. Then we compute the probability

$$p(C_1 \mid n) = p_{\alpha(1)} \prod_{j=2}^{k} \left(1 - q_{\alpha(j)}\right) + \sum_{i=2}^{k} q_{\alpha(i)} p_{\alpha(i)} \prod_{j=i+1}^{k} \left(1 - q_{\alpha(j)}\right). \tag{5}$$

This formula comes from a straightforward interpretation of priority in which we start on the right side of a name and compute the probability the name belongs to class C_1 stepwise. If $t_{\alpha(k)}$ is the rightmost token we multiple the reliability $q_{\alpha(k)}$ times the significance $p_{\alpha(k)}$ to obtain $q_{\alpha(k)} p_{\alpha(k)}$, which represents the contribution of $t_{\alpha(k)}$. The remaining or unused probability is $1 - q_{\alpha(k)}$ and this is passed to the next token to the left, $t_{\alpha(k-1)}$. The probability $1 - q_{\alpha(k)}$ is scaled by the reliability and then the significance of $t_{\alpha(k-1)}$ to obtain $(1 - q_{\alpha(k)}) q_{\alpha(k-1)} p_{\alpha(k-1)}$, which is the contribution of $t_{\alpha(k-1)}$ toward the probability that the name is of class C_1. The remaining probability is now $\left(1 - q_{\alpha(k-1)}\right)\left(1 - q_{\alpha(k)}\right)$ and this is again passed to the next token to the left, etc. At the last token on the left the reliability is not used to scale because there are no further tokens to the left and only significance $p_{\alpha(1)}$ is used.

We want to choose all the parameters p_α and q_α to maximize the probability of the data. Thus we seek to maximize

$$F = \sum_{n \in T_1} \log\left(p\left(C_1 \mid n\right)\right) + \sum_{n \in T_2} \log\left(p\left(C2 \mid n\right)\right). \tag{6}$$

Because probabilities are restricted to be in the interval $[0,1]$, it is convenient to make a change of variables through the definitions

$$p_\alpha = \frac{e^{x_\alpha}}{1 + e^{x_\alpha}},\ q_\alpha = \frac{e^{y_\alpha}}{1 + e^{y_\alpha}}. \tag{7}$$

Then it is a simple exercise to show that

$$\frac{dp_\alpha}{dx_\alpha} = p_\alpha\left(1 - p_\alpha\right),\ \frac{dq_\alpha}{dy_\alpha} = q_\alpha\left(1 - q_\alpha\right). \tag{8}$$

From (5), (6), and (8) it is straightforward to compute the gradient of F as a function of x_α and y_α and because of (8) it is most naturally expressed in terms of p_α and q_α. Before we carry out the optimization one further step is important. Let B denote the subset of $\alpha \in A$ for which all the occurrences of t_α either occur in names in T_1 or all occurrences occur in names in T_2. For any such α we set $q_\alpha = 1$ and if all occurrences of t_α are in names in T_1 we set $p_\alpha = 1$, while if all occurrences are in names in T_2 we set $p_\alpha = 0$. These choices are optimal and because of the form of (8) it is easily seen that

$$\frac{\partial F}{\partial x_\alpha} = \frac{\partial F}{\partial y_\alpha} = 0 \tag{9}$$

for such an α. Thus we may ignore all the $\alpha \in B$ in our optimization process because the values of p_α and q_α are already set optimally. We therefore carry out optimization of F using the $x_\alpha,\ y_\alpha,\ \alpha \in A - B$. For the optimization we have had good success using a Limited Memory BFGS method (Nash et al., 1991).

When the optimization of F is complete we will have estimates for all the p_α and q_α, $\alpha \in A$. We still must deal with tokens t_β that are not included among the t_α. For this purpose we train variable order Markov Models MP_1 based on the weighted set of strings $\left\{\left(t_\alpha, p_\alpha\right)\right\}_{\alpha \in A}$ and MP_2 based on $\left\{\left(t_\alpha, 1 - p_\alpha\right)\right\}_{\alpha \in A}$. Likewise we train MQ_1 based on $\left\{\left(t_\alpha, q_\alpha\right)\right\}_{\alpha \in A}$ and MQ_2 based on $\left\{\left(t_\alpha, 1 - q_\alpha\right)\right\}_{\alpha \in A}$. Then if we allow $mp_i\left(t_\beta\right)$ to represent the prediction from model MP_i and $mq_i\left(t_\beta\right)$ that from model MQ_i, we set

$$p_\beta = \frac{mp_1(t_\beta)}{mp_1(t_\beta)+mp_2(t_\beta)},\ q_\beta = \frac{mq_1(t_\beta)}{mq_1(t_\beta)+mq_2(t_\beta)} \quad (10)$$

This allows us to apply the priority model to any name to predict its classification based on equation 5.

4 Results

We ran all three methods on the SemCat sets *gp1*, *gp2* and *gp3*. Results are shown in Table 2. For evaluation we applied the standard information retrieval measures precision, recall and F-measure.

$$precision = \frac{rel\_ret}{(rel\_ret + non - rel\_ret)}$$

$$recall = \frac{rel\_ret}{(rel\_ret + rel\_not\_ret)}$$

$$F - measure = \frac{2 * precision * recall}{(precision + recall)}$$

For name classification, *rel_ret* refers to true positive entities, *non-rel_ret* to false positive entities and *rel_ not_ret* to false negative entities.

Table 2. Three-fold cross validation results. P = Precision, R = Recall, F = F-measure. PCFG = Probabilistic Context-Free Grammar, LM = Bigram Model with Witten-Bell smoothing, PM = Priority Model.

Method	Run	P	R	F
PCFG-3	gp1	0.883	0.934	0.908
	gp2	0.882	0.937	0.909
	gp3	0.877	0.936	0.906
PCFG-8	gp1	0.939	0.966	0.952
	gp2	0.938	0.967	0.952
	gp3	0.939	0.966	0.952
LM	gp1	0.920	0.968	0.944
	gp2	0.923	0.968	0.945
	gp3	0.917	0.971	0.943
PM	gp1	0.949	0.968	0.958
	gp2	0.950	0.968	0.960
	gp3	0.950	0.967	0.958

5 Discussion

Using a variable order Markov model for strings improved the results for all methods (results not shown). The *gp1-3* results are similar within each method, yet it is clear that the overall performance of these methods is PM > PCFG-8 > LM > PCFG-3. The very large size of the database and the very uniform results obtained over the three independent random splits of the data support this conclusion.

The improvement of PCFG-8 over PCFG-3 can be attributed to the considerable ambiguity in this domain. Since there are many cases of term overlap in the training data, a grammar incorporating some of this ambiguity should outperform one that does not. In PCFG-8, additional production rules allow phrases beginning as CATPs to be overall NotCATPs, and vice versa.

The Priority Model outperformed all other methods using F-measure. This supports our impression that the right-most words in a name should be given higher priority when classifying names. A decrease in performance for the model is expected when applying this model to the named entity extraction (NER) task, since the model is based on terminology alone and not on the surrounding natural language text. In our classification experiments, there is no context, so disambiguation is not an issue. However, the application of our model to NER will require addressing this problem.

SemCat has not been tested for accuracy, but we retain a set of manually-assigned scores that attest to the reliability of each contributing list of terms. Table 2 indicates that good results can be obtained even with noisy training data.

6 Conclusion

In this paper, we have concentrated on the information inherent in gene and protein names versus other biomedical entities. We have demonstrated the utility of the SemCat database in training probabilistic methods for gene and protein entity classification. We have also introduced a new model for named entity prediction that prioritizes the contribution of words towards the right end of terms. The Priority Model shows promise in the domain of gene and protein name classification. We plan to apply the Priority Model, along with appropriate contextual and meta-level information, to gene and protein named entity recognition in future work. We intend to make SemCat freely available.

Acknowledgements

This research was supported in part by the Intramural Research Program of the NIH, National Library of Medicine.

References

T. L. Booth. 1969. Probabilistic representation of formal languages. In: *IEEE Conference Record of the 1969 Tenth Annual Symposium on Switching and Automata Theory*, 74-81.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. *Technical Report TR-10-98*, Computer Science Group, Harvard University.

Eugene Charniak. 1993. *Statistical Language Learning*. The MIT Press, Cambridge, Massachusetts.

K. Bretonnel Cohen, Lynne Fox, Philip V. Ogren and Lawrence Hunter. 2005. Corpus design for biomedical natural language processing. *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*, 38-45.

The Gene Ontology Consortium. 2000. Gene Ontology: tool for the unification of biology, *Nat Genet*. 25: 25-29.

Henk Harkema, Robert Gaizauskas, Mark Hepple, Angus Roberts, Ian Roberts, Neil Davis and Yikun Guo. 2004. A large scale terminology resource for biomedical text processing. *Proc BioLINK 2004*, 53-60.

Vasileios Hatzivassiloglou, Pablo A. Duboué and Andrey Rzhetsky. 2001. Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics* 17 Suppl 1:S97-106.

J.-D. Kim, Tomoko Ohta, Yuka Tateisi and Jun-ichi Tsujii. 2003. GENIA corpus--semantically annotated corpus for bio-textmining. *Bioinformatics* 19 Suppl 1:i180-2.

Donald A. Lindberg, Betsy L. Humphreys and Alexa T. McCray. 1993. The Unified Medical Language System. *Methods Inf Med* 32(4):281-91.

Hongfang Liu, Stephen B. Johnson, and Carol Friedman. 2002. Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. *J Am Med Inform Assoc* 9(6): 621–636.

Donna Maglott, Jim Ostell, Kim D. Pruitt and Tatiana Tatusova. 2005. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res.* 33:D54-8.

Alexa T. McCray. 1989. The UMLS semantic network. In: Kingsland LC (ed). *Proc 13rd Annu Symp Comput Appl Med Care*. Washington, DC: IEEE Computer Society Press, 503-7.

Ryan McDonald and Fernando Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics* 6 Supp 1:S6.

S. Nash and J. Nocedal. 1991. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization, *SIAM J. Optimization*1(3): 358-372.

Goran Nenadic, Irena Spasic and Sophia Ananiadou. 2003. Terminology-driven mining of biomedical literature. *Bioinformatics* 19:8, 938-943.

Raf M. Podowski, John G. Cleary, Nicholas T. Goncharoff, Gregory Amoutzias and William S. Hayes. 2004. AZuRE, a scalable system for automated term disambiguation of gene and protein Names *IEEE Computer Society Bioinformatics Conference*, 415-424.

Lawrence H. Smith, Lorraine Tanabe, Thomas C. Rindflesch and W. John Wilbur. 2005. MedTag: A collection of biomedical annotations. *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*, 32-37.

Lorraine Tanabe, Natalie Xie, Lynne H. Thom, Wayne Matten and W. John Wilbur. 2005. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics* 6 Suppl 1:S3.

I. Witten and T. Bell, 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory* 37(4).

Alexander Yeh, Alexander Morgan, Mark Colosimo and Lynette Hirschman. 2005. BioCreAtIvE Task 1A: gene mention finding evaluation. *BMC Bioinformatics* 6 Suppl 1:S2.

Hong Yu, Carol Friedman, Andrey Rhzetsky and Pauline Kra. 1999. Representing genomic knowledge in the UMLS semantic network. *Proc AMIA Symp*. 181-5.